Mobile Edge Cooperation Optimization for Wearable Internet of Things: A Network Representation-based Framework

Xiangjie Kong, *Senior Member, IEEE,* Shiqin Tong, Haoran Gao, Guojiang Shen, Kailai Wang, Mario Collotta, *Member, IEEE,* Ilsun You, *Senior Member, IEEE,* Sajal K. Das, *Fellow, IEEE*

***Abstract*—As a new computing paradigm, edge computing emerges in various fields. Many tasks previously relied on cloud computing are distributed to various edge devices which cooperate to complete the tasks. However, circumstantial factors in the edge network (e.g., functionality, transmission efficiency, resource limitation) become more complex than those in cloud computing. Consequently, there is instability that cannot be ignored in the cooperation between the edge devices. In this paper, we propose a novel framework to optimize edge cooperative network (ECN), called ECN-Opt, to improve the performance of edge computing tasks. Specifically, we first define the evaluation metrics for cooperation. Next, the cooperation of ECN is optimized to improve the performance of specific tasks. Extensive experiments using real datasets from wearable sensors on the players in soccer teams, demonstrate that our ECN-Opt framework performs well. And it also validate the effectiveness of the proposed optimization algorithm.**

***Index Terms*—Mobile Edge Computing, Wearable Sensor, Cooperative Optimization, Edge Cooperative Network.**

I. Introduction

WEARABLE sensors have been widely used in the Industrial Internet of Things (IIoT) setting, mainly as data producers [1]. Moreover, a considerable number of wearable sensors have enough computing power to be able to consume data in the edge network. As a new computing paradigm, edge computing is emerging in various fields, such as internet of vehicles (IoV), big data analytics, distributed deep learning, and so on. Edge computing also provides better security and privacy compared with traditional cloud computing. In the development of edge computing, three main architectures have been proposed: namely cloudlets [2], fog computing [3], and mobile edge computing (MEC) [4], [5]. In particular, the MEC architecture (shown in Fig. 1) has attracted considerable attention both in academia and industry. Indeed, since the cloudlets framework originally proposed in 2009, the devices performing computation-intensive tasks gradually moved from the cloud to the edge, because edge computing significantly improves the quality of service (QoS).

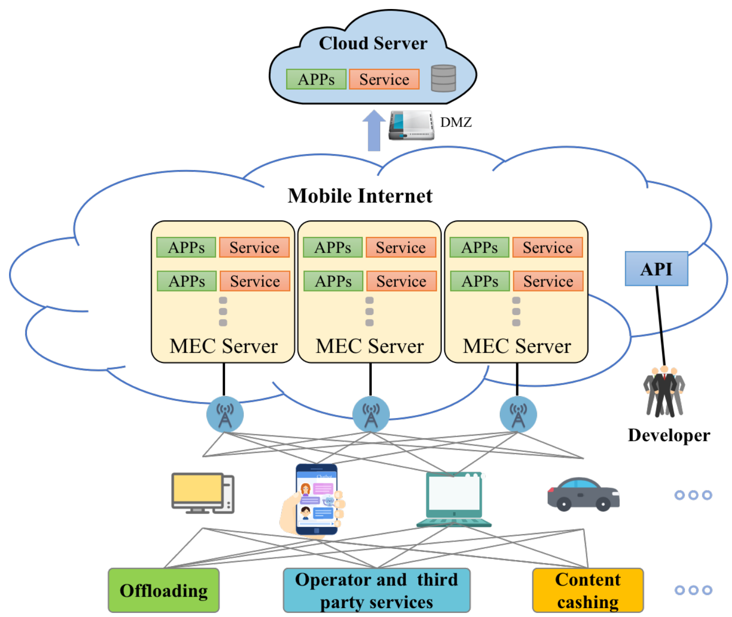


Fig. 1. Architecture and applications of MEC.

With the emergence of a wide variety of edge computing applications, the management of edge resources and the cooperation between the edge nodes pose new challenges and opportunities. In other words, cooperation among tasks in edge computing architectures need to be optimized according to the instability of the edge environments. To this end, there exist several approaches in the literature. For example, Ning et al. [6] propose a method to promote AI heuristic computing, which can cache and communicate resources to the vicinity of intelligent vehicles, thereby jointly implementing RSU peer-to-peer offloading in the IoV framework. Their research has achieved outstanding results in minimizing the total network delay. Qu et al. [7] propose a solution for cooperative caching for multiple bit rate videos. Funai et al. [8] propose a heuristic algorithm for iterative task assignment to optimize the assignment of computational tasks in a multi-hop cooperative network.

However, most of the existing work on edge cooperation optimization is limited to a particular task, and there is no

The work of S. K. Das is partially supported by the US National Science Foundation under NSF grant numbers CNS-2008878 and CCF-1725755.
(Corresponding author: Guojiang Shen.)

X. Kong and G. Shen are with College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: xjkong@ieee.org; gjshen1975@zjut.edu.cn).

S. Tong, H. Gao and K. Wang are with School of Software, Dalian University of Technology, Dalian 116620, China (e-mail: shiqin.tong@outlook.com; damiangao96@gmail.com; kailai.w@outlook.com).

M. Collotta is with the Faculty of Engineering and Architecture, Kore University of Enna, Enna 94100, Italy (e-mail: mario.collotta@unikore.it).

I. You is with Department of Information Security Engineering, Soonchunhyang University, Asan-si 31538, Korea (e-mail: ilsunu@gmail.com).

S. K. Das is with the Department of Computer Science, Missouri University of Science and Technology, Rolla, MO 65409, USA (e-mail: sdas@mst.edu).

general way to evaluate and optimize edge cooperation. Our motivation particularly stems from the cooperative evaluation and optimization of sports. We envision that the evaluation and optimization of cooperation between edge computing and sports teams have a lot of similarity. This is because the analysis of cooperation of sports team has the following characteristics: (i) the available resources are limited and adjustments to the current allocation are also restricted; (ii) the objective environment is unfavorable and has many uncertainties (eg., opponent's interference); (iii) the overall goal is clear; (iv) each node plays a different role and their abilities are different; and (v) the data source is the same as that of the node cooperative network (e.g., IoT sensors).

Recently, some interesting work has been proposed on the evaluation of personal contribution to the sports competition. Decroos et al. [9] propose a novel data-driven framework for valuing actions in a soccer game. In their research, individual contributions are evaluated further. Liu et al. [10] apply deep reinforcement learning in ice hockey for context-aware player evaluation. In [11], the contributions of soccer players to chance creation are measured by valuing their passes. Other studies focus on decision making and collaborative optimization. Fernandez et al. [12] propose an expected possession value framework for soccer based on deep learning. In their research, decision making is discussed at both the team and player level. Bouchekara et al. [13] propose a novel optimization algorithm, the Most Valuable Player Algorithm (MVPA), that considers the goals of individuals and the team. Considering that both node cooperation in edge computing and team cooperation in competitive sports have a network structure, we investigate this problem from the viewpoint of network representation.

Network representation learning aims to extract potential information by representing the nodes in network as a low-dimensional, dense vector form. These vectors can be used for information inference in the latent space. Network representation learning usually has two advantages: (i) the representation learned in the low-dimensional space can reconstruct the original network structure; and (ii) the learned representation can support network information inference. Existing methods are divided into three categories: matrix factorization, random walk, and deep neural networks. Matrix factorization based methods [14] typically use two steps: construction of relationship matrix between the nodes, and deriving the network representation vector by utilizing matrix decomposition operation. For random walk-based methods [15], the idea is to correspond nodes to the words in Natural Language Processing (NLP). On the other hand, [16] algorithm based on the neural networks uses a function learning model suitable for fitting highly nonlinear structures to characterize the network.

In this paper, we propose a novel framework, call ECN-Opt, to optimize the ECN with edge computing. By exploring the cooperation between nodes, we can quantitatively analyze the structure of the network and improve the performance of collaborative tasks. First, we measure the amount of interaction between the edge nodes to establish an ECN. Second, a network representation algorithm is used to represent each node in vector. Then we design algorithms to explore the cooperative mode between the nodes. Based on micro-cooperation characteristics, we evaluate individual nodes and propose the evaluation indicators of ECN. Finally, we propose a method to optimize the node cooperative network.

Our contributions are summarized as follows:

- We define an ECN based on the edge tasks. We focus on the cooperation relationship between the edge nodes, explored by the network representation algorithm.
- We develop a novel optimization framework for ECN, called ECN-Opt that includes micro-cooperation modes extracted from the results of network representation, Based on this, macro-cooperation evaluation indicators and replacement node recommendations can be realized.
- We select the novel sports cooperation network based on wearable sensors of players to verify our indicators, and design experiments. Results demonstrate that our cooperative evaluation indicators are pervasive and applicable to a variety of edge network tasks. We use real datasets to analyze the optimization of the sports cooperation network.

The rest of this paper is organized as follows. Section II reviews the related work about edge computing, cooperation, and representation learning algorithms. Section III explains our proposed approach in detail. Following that, we conduct extensive comparative experiments to verify the effectiveness of our approach in Sections IV and V. Finally, we conclude the paper in Section VI.

II. Related Work

In this section, we review state of the art literature on pervasive edge computing, cooperation and representation learning.

A. Edge Computing

The emergence of edge computing paradigm has enabled to explore the potential of IoT because the computing can now be performed on the edge devices, thereby reducing the response time and guaranteeing privacy. The edge of the network is also changing from data consumer to data producer as well as data consumer [17]. Although the edge computing technology is being widely used in various applications, due to limited computing resources of terminal devices, some or all of the computing tasks are often offloaded to the cloud from the MEC nodes in order to save power (energy) of the mobile devices [18]. For an example, the face detection system in mobile phones system makes full use of the computing capacity of edge devices [19]. The MEC server can use of edge devices to track information about the end user and consequently make inference to provide context-aware services [20], [21]. The high-quality computing capability provided by edge computing and the huge data services by IoT not only enrich the edge applications, but also put forward significant challenges for cooperation between the edge devices (nodes).

B. Cooperation of Edge Devices

In recent years, a lot efforts have been made on the cooperation of edge devices. Yan et al. [22] use federated learning to distribute the edge resources. Leyva et al. [23] propose a network coding cooperation (NCC) protocol for efficient large-scale content delivery, and provide an analytical model to describe its behavior. Given the current research focus is mainly on team cooperation networks, a loyalty program (LP) is proposed in [24] to effectively promote cooperation, while in [25], team metrics is proposed to analyze the significance of team formation. In this paper, we analyze and optimize the cooperation of the edge network based on the existing work on team network analysis.

C. Representation Learning

Network representation technology plays an important role in the analysis of network. With many edge tasks abstracted into networks, the network representation can be utilized to learn about networks and gather valuable information. The basic idea of network representation is to map the network to a low-dimensional vector space, and then use a low-dimensional dense vector to represent nodes in the network. Perozzi et al. [15] propose DeepWalk that utilizes random walk in networks to stimulate sentences in language models, so that the network representation problem is transformed into the word representation problem. Random walk has been a general tool for learning network embedding, such as node2vec [26] which controls random walk more subtly. In [27], the authors propose LINE to learn network representations via preserving the first and second order pairwise proximity. Compared with models that only consider node features or graph structures, GNN has become a broad solution. For an example, Zhu et al. [28] propose a new graph convolution operator. They call this framework bilinear graph neural network (BGNN). BGNN augments the weighted sum with pairwise interactions of the representations of neighbor nodes. But with the outstanding effect is the complexity of the algorithm, a lot of existing research is focused on simplifying the algorithm. For example, He et al. [29] propose a new model called LightGCN, which contains only the most important component of GCN: neighborhood aggregation. However, the algorithms such as GNN and GCN are not suitable for the edge network due to the high algorithm complexity. Therefore, we choose Deepwalk [15] with low algorithm complexity as part of our framework. In addition, for edge tasks, we believe that only the first-order relationship of nodes needs to be considered, and the potential second-order relationship between nodes can be ignored. Deepwalk [15] has neglected the second-order relationship between nodes, which is suitable for edge networks. Based on Deepwalk [15], our framework can explore the cooperative relationship among nodes to optimize the network in the case of limited resources in the edge network. Therefore, the performance of edge tasks is improved.

III. Methodologies

In this section, we first present definitions. And then, an overview of our framework is introduced. Finally, a detailed description of each component is listed.

Definition 1 (Edge Cooperative Network (ECN)): Resource dilemma has always existed in edge networks, such as limited computing power of nodes. In this environment, ECN refers to the intensity of cooperation between edge nodes within a certain period of time. It is defined as $G = (V, E, W)$. $V = \{v_1, v_2, ..., v_N\}$ is the set of edge nodes. $E = \{e_{i,j}\}$ indicates the set of cooperation relationships between v_i and v_j. W is weight matrix of the network, and $w_{i,j}$ controls the strength of the cooperation $e_{i,j}$.

Definition 2 (Edge Node Representation): Network representation method is applied to learn low-dimensional representations of vertex in ECN. Namely, ECN is embedded into the same d-dimensional space, and each node is represented by a d-dimensional vector $\Phi(i) \in \mathbb{R}^{|V|\times d}$, where $d \ll |V|$.

Definition 3 (Cooperative Mode): The tightness of cooperation between nodes is used by us to define the cooperative mode of nodes. It can be calculated by the distance of nodes in the representation space, denoted by $D_{i,j} = \|\Phi(i) - \Phi(j)\|$, where $\|\cdot\|$ means Euclidean distance. If $D_{i,j}$ is less than a threshold c, we assume that they have a strong cooperation. In addition, the cooperative mode among nodes also needs to consider complex factors, including the antagonistic relationship and the function of the nodes. Based on this, the cooperative relationship of the nodes in the ECN is divided into two types of relationship modes: binary models and ternary models. In particular, for a node, it has only one pair of strong relationships is a binary model. Given three nodes, if each pair of nodes has a strong cooperation, it is called ternary models in this paper. The binary model has the advantages of flexibility, but when facing important tasks, the triangle motif has a stronger advantage.

A. An Overview of the Framework

The framework, ECN-Opt, is divided into three parts, the first part is the establishment of ECN, and the second part is to apply the network representation algorithm to the ECN to obtain the low-dimensional representation of nodes. In the third part, we first identify the cooperative mode of nodes. Then we evaluate the nodes and optimize our network. Fig. 2 shows the details of the framework.

B. Establishment of ECN

The edge network equipment has fewer resources, so when we consider the related edge network tasks, we should pay attention to the difference between edge network tasks and ordinary tasks. For an example, the computing power dominated by tasks in the edge network is less than ordinary tasks. In addition, in edge cooperative tasks, nodes are usually different from each other. Differences including function, computing ability, resource, etc. The differences between nodes make it difficult to optimize the cooperation by controlling all status of nodes precisely. A real-world ECN example is shown in Fig. 3.

For the foregoing reasons, in our research, the edge cooperation task is abstracted into an ECN in this paper. We consider the role of nodes. Intuitively, the role of nodes are implicit in the interaction between nodes. For the establishment of network, first of all, We treat base stations and wearable devices equally and define them as equivalent nodes in the network.

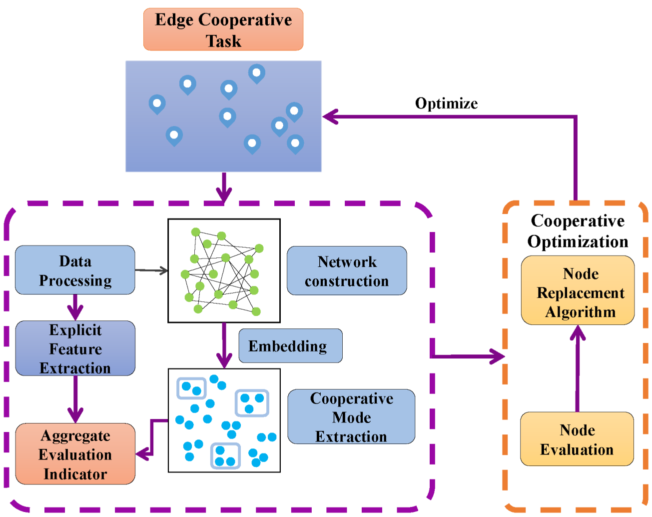


Fig. 2. Detailed process of the edge cooperation optimization framework, ECN-Opt. We abstract the task as an ECN and extract the cooperative mode from the ECN. On the basis of this, we establish cooperation evaluation indicators and optimize cooperative network.

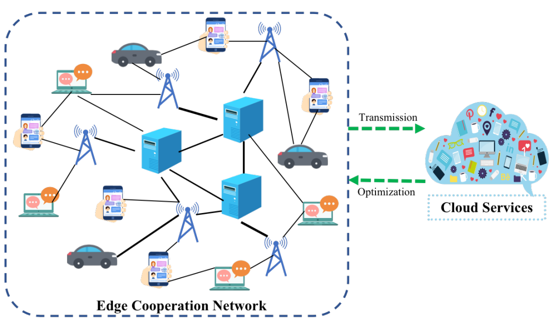


Fig. 3. In ECN, we treat the cooperation between devices as edges. Thicker edges represent stronger cooperations. ECN can interact with cloud services.

Then, for the tasks, the records of interactions between nodes, denoted as $\{\delta_{from,to} = (v_{from}, v_{to})|\delta \in \Delta\}$. A way to calculate the edge weights is $w_{i,j} = \sum^{\Delta} \delta_{i,j}$. The steps are shown in Algorithm 1. Specifically, we take the devices in the edge network as the nodes of the network, and regard the interaction behavior between the devices of the edge network as the weight of the edges between the nodes.

C. Cooperative Mode Extraction

However, with the establishment of ECN, the Deepwalk [15] algorithm is used to embed the structure of the network. The mapping function is denoted as $\Phi(\cdot)$, which maps vertex $v_i \in V$ to $\mathbb{R}^{|V|\times d}$. Deepwalk adopts a random walk strategy to generate sequences of nodes of fixed length t. Then these sequences are considered equivalent to sentences in language model. In random walk phase, the next node in the sequence is selected with the probability $p_{i,j}$:

$$p_{i,j} = \frac{w_{i,j}}{\sum_{k\in N_i} w_{i,k}} \tag{1}$$

In the equation above, N_i refers to all the neighbors of i. To learn low-dimensional representations of vertices, a Skip-gram language model used in [26] is applied with nodes sequences

Algorithm 1 Establishment of ECN

Input: Nodes interaction events C_e, number of nodes n
Output: ECN G

init G
for $i, j < n$ **do**
  $G[i][j] = 0$ // Initialize operation
end for
for each event $e \in C_e$ **do**
  $G[e.from][e.to] = G[e.from][e.to] + 1$
  // Count events as the weight of the edge
end for
return G // The network is established

as input sentences as shown in Fig. 4. The objective of the Skip-gram model is maximize the average log probability:

$$\max \frac{1}{N} \sum_{i=1}^{N} \sum_{-\omega \le j \le \omega, j \ne 0} \log p(v_{i+j}|v_i) \tag{2}$$

where ω denotes the window size. Optimizing this goal

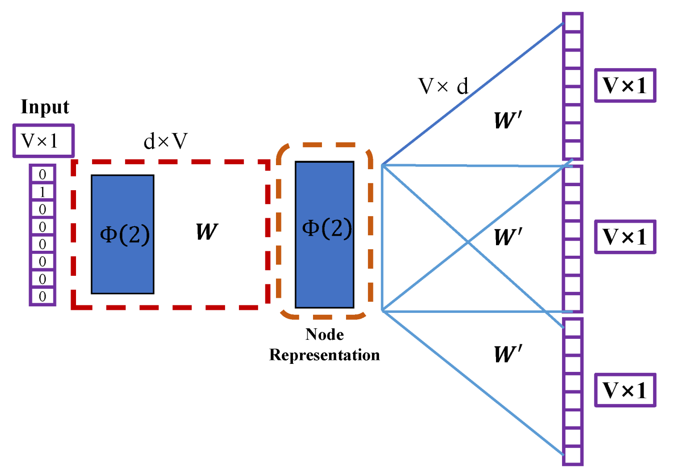


Fig. 4. Skin-Gram proposes a way to train the context of the network node when the long sequence is already known.

requires high computational cost. Thus, negative sampling is used to approximate the optimization of the objective function.

$$\begin{aligned} \min_{\Phi} - \sum_{j=0, j\ne\omega}^{2\omega} \Phi(i-\omega+j)\cdot\Phi(i) \\ + 2\omega \log \sum_{k=1}^{|V|} \exp\{\Phi(k)\cdot\Phi(i)\} \end{aligned} \tag{3}$$

We calculate the node's cooperative mode based on the node's latent vector representation. As described in Definition 3, the cooperative mode between nodes can be referred to the distance of latent vector between nodes in the representation space. we measure the distance between the vectors in the latent space through the Euclidean distance formula.

$$d_{x,y} = \|\Phi(x) - \Phi(y)\| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{4}$$

Algorithm 2 Cooperative Mode Extraction

Input: The representation of each node Emb, threshold c
Output: Binary mode count τ_2; Ternary mode count τ_3

init $\tau_2 = 0$, $\tau_3 = 0$, dis // Distance matrix
init set $binarys$, $ternarys$
$n = Emb.length$
for $i, j < n$ **do**
  $dis[i][j] = \|Emb[i] - Emb[j]\|$ // Distance in vector space
end for
$\theta = c * dis.avg$
for $i, j > n$ **do**
  if $dis[i][j] \geq \theta$ **then**
    $binarys.append((i, j))$ // Binary model
  end if
end for
for $k < n$, $(i, j) \in binarys$ **do**
  if $dis[i][k] \leq \theta$ **and** $dis[j][k] \leq \theta$ **then**
    $ternarys.append((i, j, k))$ // Ternary model
  end if
end for
$\tau_2 = binarys.size$, $\tau_3 = ternarys.size$
return τ_2, τ_3

The cooperative mode extraction algorithm is proposed as Algorithm 2. We count the number of binary and ternary cooperation modes, respectively, denoted as τ_2 and τ_3. The cooperative mode between the nodes is strong with small spacing. On the contrary, the cooperative relationship between the nodes with large spacing is relatively weak. We distinguish strong and weak cooperative relationships between nodes by specifying the cooperation strength threshold c. In this paper, we only discuss the impact of strong cooperation within ECN. It has been observed that strong cooperation exhibits a more centralized cooperative configuration in cooperative network. Strong relationships themselves are a form of binary model. When ternary strong relationships occur frequently, it means that cooperation is not evenly distributed, but concentrated on certain nodes.

D. Evaluation and Optimization of Cooperative Network

1) Node Evaluation: In ECN, multiple types of nodes may be required to cooperate well to complete a task. Based on this, we explore the characteristics of nodes to measure the cooperative ability of nodes. Intuitively, all edge nodes participating in cooperation have the following three characteristics: cooperation mode preference M_{node}, and interaction frequency I, task contribution T. The cooperation preference can be obtained simply by $M_{node} = \frac{\tau_3}{\tau_2}$. The interaction frequency I can be defined as the ratio of the node interactions to the total network interactions within the time interval t. Therefore, the interaction frequency I_i of individual node v_i is formulated as follows:

$$I_i = T \frac{\sum_{e_{i,j} \in E} w_{i,j}}{t}, \tag{5}$$

In addition, task contribution T is closely related to specific tasks. By incorporating these three features into a vector, we can calculate the similarity of each pair of vectors. Accordingly, individual nodes are effectively evaluated.

2) Network Evaluation: The evaluation of cooperative network depends on three aspects: frequency of cooperation in the network Ω, cooperative mode preference M_{ECN}, and the constraints of the objective environment Ψ. Ω can usually be calculated by the number of interactions between nodes in an ECN. There appears to be an obvious correlation between the frequency of cooperation and the quality of cooperation. In the case of a limited number of nodes, high frequency of cooperation represents the close working relationship between nodes. In addition to the characteristics of ECN, the impact of other factors should not be ignored. For an example, network fluctuations affect cooperation so that it is unfair to evaluate the quality of cooperation by ignoring such interference. Intuitively, from the macro perspective of the ECN, we believe that the impact of factors such as network fluctuations is reflected in the network's macro cooperative mode preference M_{ECN}. Additionally, the quality of cooperation is limited when external environmental factors Ψ are taken into account. Based on the above reasons, we propose a pervasive cooperative quality model, called Quality of Cooperation (QoC):

$$QoC = \frac{(M_{ECN})^{\mu} \times \Omega}{\Psi} \tag{6}$$

where μ is the parameter that controls the preference of cooperative mode. Ω and Ψ two variables are task-specific. QoC effectively measures the quality of network cooperation through the number of interactions between edge nodes. In the experimental part, we introduce the application of QoC to specific problems.

3) Optimization of Edge Cooperation: In general, nodes tend to cooperate with their own similar nodes. For an example, among the nodes that have a cooperative relationship with the data transmission node, 80% of the nodes function as data transmission, and 20% of the nodes function as others. Therefore, the node's cooperative mode preference M_{node} can also be regarded as the similarity of node functions. In the edge cooperation task, we find that after replacing similar nodes with weak capabilities or reducing unnecessary similar nodes, the M_{node} and M_{ECN} are effectively improved, and the performance of the network task is significantly improved. Therefore, we explore the M_{node} to find the nodes that can be optimized in the network. For the improvement of M_{node} index, We first use DPCA [30] to perform density-based clustering on structural information. It calculates the local density value of all sample points and the distance to the high local density point, which are given as:

$$\rho_i = \sum_{i=1} \chi(d_{i,j} - d_c) \tag{7}$$

$$\delta_i = \min_{j:\rho_j > \rho_i} d_{i,j} \tag{8}$$

where ρ_i denotes the number of nodes whose distance with v_i is greater than cut-off distance d_c. Based on these values, a decision map is obtained. Then we select the appropriate sample points on the decision graph as the cluster center for clustering. According to the above results, we find that there are two types of nodes: (i) Hub nodes which have a large number of similar nodes. (ii) Separate nodes which have a small number of similar nodes. In order to make replacement of nodes rewarding, we first calculate the distance from each node to the cluster center closest to it, and then calculate the number of nodes at the same distance to measure the number of similar nodes S of each node. When the S of a node is greater than 10% of the number of nodes in the network, the node is a hub node within the cluster, otherwise, it is a discrete node. Finally, we sort the two types of nodes according to the number of similar nodes. By sorting, we determine the nodes that need to be replaced and optimized. Moreover, it should be noted that we think that discrete nodes have a higher priority than hub nodes. Specifically, we first replace nodes by rank in the category of discrete nodes, and then replace nodes by rank in the category of hub nodes.

IV. Experiments

For experiments, we innovatively select the sports cooperation network based on wearable sensors of players to verify our framework. In this section, we first introduce the selected real dataset. Then we compare evaluation metrics of cooperation network with other models. Finally, the node replacement events in the real dataset are analyzed accordingly. Based on this, we prove the effectiveness of the proposed framework.

A. Dataset Selection and Description

In a soccer game, in order to effectively analyze the player's cooperation and the state of the team, the coach will make the player wear sensors and obtain the player's dynamic data including position, acceleration, etc. The player network is a network established with players as nodes and the number of activities between players as the weight of edges. A simple figure is shown in Fig. 5. The European Cup (2008) dataset collected by the edge sensor on the players is used by us to analyze the cooperation between players. It contains the data covering 23,429 passes between 366 players (30 target team players, and 336 players from opposing teams), and 59,271 game events. Events between players include but are not limited to passing, scoring, etc. We selecte ball possession and passes in the dataset for visualization. As shown in Fig. 6, there appears to be a strong correlation between the trends of ball possession and the number of passes.

B. Comparison of Cooperative Evaluation

In the evaluation part of our framework, we verify the effectiveness of the cooperative network evaluation by comparing with other models.

a) The Quality of Cooperation (QoC): Specific to the soccer team cooperation, QoC also shows strong adaptability. The interaction events of edge nodes can be naturally transfered to the passing events of soccer. For environmental factors, it is innovative to take into account the performance of the opponents, that is, $\Omega = \sum_i \sum_j w_{i,j}$ and $\Psi = \left(\frac{\tau_3^{(opp)}}{\tau_2^{(opp)}}\right)^{\mu} \times \Omega^{(opp)}$. Finally, QoC is derived as follow.

$$QoC = \left(\frac{\tau_3}{\tau_2} \times \frac{\tau_2^{(opp)}}{\tau_3^{(opp)}}\right)^{\mu} \times \frac{\sum_i \sum_j w_{i,j}}{\sum_i \sum_j w_{i,j}^{(opp)}} \tag{9}$$

b) Pezzali Score: Cintia et al. [25] propose Pezzali Score to evaluate the team's offensive efficiency and defensive efficiency. Inspired by the thought that good defense is as important as good offense, Pezzali Score uses the ratio of goals to shots made by the team as the offensive efficiency while the reciprocal of the opponent's offensive efficiency as defensive efficiency. Pezzali Score is defined as:

$$PezzaliScore = \frac{|goals(team)|}{|attempts(team)|} \times \frac{|attempts(opponent)|}{|goals(opponent)|} \tag{10}$$

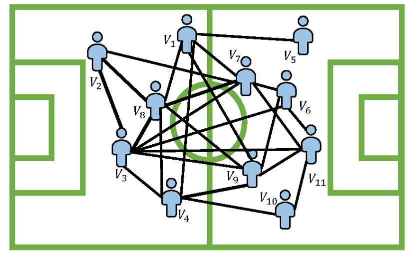
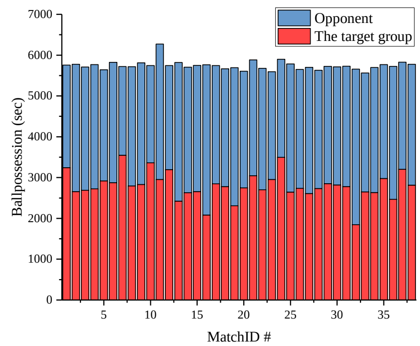
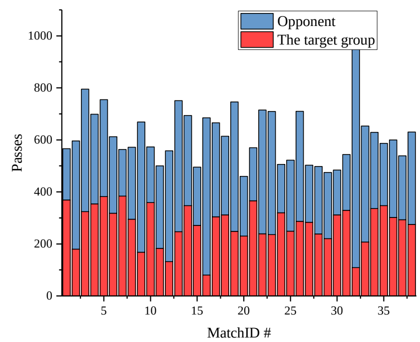


Fig. 5. The ECN established with the soccer game dataset. Edge represents a large number of interaction events between players. Edges have weights.



(a) ballpossession



(b) the number of passes

Fig. 6. Ball possession represents the purpose of the cooperation network, and the number of passes indicates the occurrence of cooperation. By visualizing the data, we find that the changing trends of ball possession and the number of passes have a strong correlation.

c) Playing Style: The club increasingly considers players' style during recruitment to determine the most suitable players for its team. The style of the players includes their habits and abilities in the game. In this study, we use traditional indicators to measure the player's personal style. The player is measured by his average goals per second, and the team is measured by its average number of goals per second, whose equation is as follows:

$$PS = \frac{Num_{activities}}{Time_{unit}} \tag{11}$$

Finally, we calculate the average ($PS(mean)$) and variance ($PS(var)$) of the PS for each game.

C. Optimization of ECN

In fact, whether ECN is optimized can be judged by network evaluation indicator. Based on our evaluation indicator, we analyze the change of the indicator value at each time stage during a game to verify the optimization performance of our framework. Meanwhile, the replacement of players plays an important role in the optimization of ECN. For example, replacing nodes with poor interaction capabilities in the network can effectively improve the efficiency. Consequently, we obtain the players that are recommended for replacement in each game with our framework. Besides, we observe the actual replacement of players in a game. By comparing the consistency and the difference between the recommended and actual replacement, we analysis the evaluation metrics change to show the effectiveness of optimization operation in ECN. In this part of the experiment, a number of typical cases are specifically analyzed.

V. Performance and Discussion

This section discusses the experimental results in detail. Experimental results and analysis finally prove the effectiveness of this framework.

A. Cooperative Evaluation Result

We calculate our cooperative evaluation indicator QoC and all baseline metrics for the selected teams in 33 European Cup matches in 2008. As the Ball Possession Rate (BPR) can reflect the performance of teams, BPR of the target team and all baseline metrics are plotted in the Fig. 7. Obviously, our metrics QoC fits well with BPR, as shown in Fig. 7(d). When the team performs well (the BPR is higher), QoC value is also extremely high. The fitting of other baseline metrics and BPR are inferior to that of QoC, which is displayed from Fig. 7(a) to the Fig. 7(c).

B. Correlation between Cooperative Metrics and Goals

In order to test the effectiveness of the cooperation evaluation metrics, in this paper, we compare our framework with the baseline on a variety of correlation metrics.

- Pearson Correlation Coefficient (PCC). Pearson correlation coefficient is widely used to measure the correlation between two kinds of data. When the value is positive, the data is positively correlated, otherwise, it is negatively correlated. The absolute value represents the strength of the correlation. The formula is as follows:

$$PCC = \frac{N\sum XY - \sum X \sum Y}{\sqrt{N\sum X^2 - (\sum X)^2}\sqrt{N\sum Y^2 - (\sum Y)^2}} \tag{12}$$

- Spearman's Rank Correlation Coefficient (SRCC). It is a common measure of the rank correlation of two variables. It uses monototic function to evaluate the correlation between variables. With no duplicate values in the data, the spearman correlation coefficient is $+1$ or -1 when the two variables are completely monotonically correlated. SRCC is defined as:

$$SRCC = \frac{cov(rg_X, rg_Y)}{\sigma_{rg_X}\sigma_{rg_Y}} \tag{13}$$

where $cov(\cdot,\cdot)$ is used to compute the covariance, and rg denotes rank.

- Cosine Similarity (CS). The cosine similarity of two vectors is evaluated by computing the cosine of the angle between them. The equation is as follows:

$$cos(\theta) = \frac{a \cdot b}{\|a\| \times \|b\|} \tag{14}$$

TABLE I
The Correlation Metrics Comparison

Indicator	PCC	SRCC	CS
QoC (ours)	**0.7009**	**0.9983**	0.9109
PS (mean)	0.0567	-0.0175	0.9432
PS (var)	0.3127	-17.8046	**0.9564**
Pezzali Score	-0.0930	0.9828	0.6550

Table I shows the correlation between the metrics and the BPR differences. Experiments show that our metrics have excellent performance. QoC has the best performance on PCC. In SRCC, QoC and Pezzali Score have similar results, and QoC is slightly better. But in terms of CS, Playing Style including PS (mean) and PS (var) is better. Generally speaking, QoC has a significant correlation with the team's BPR, and our metrics can accurately describe team performance. Therefore, it is reasonable to believe that our metrics QoC also have a superior performance in evaluating the cooperative quality of edge nodes.

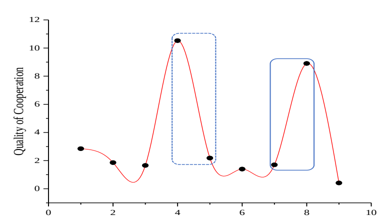
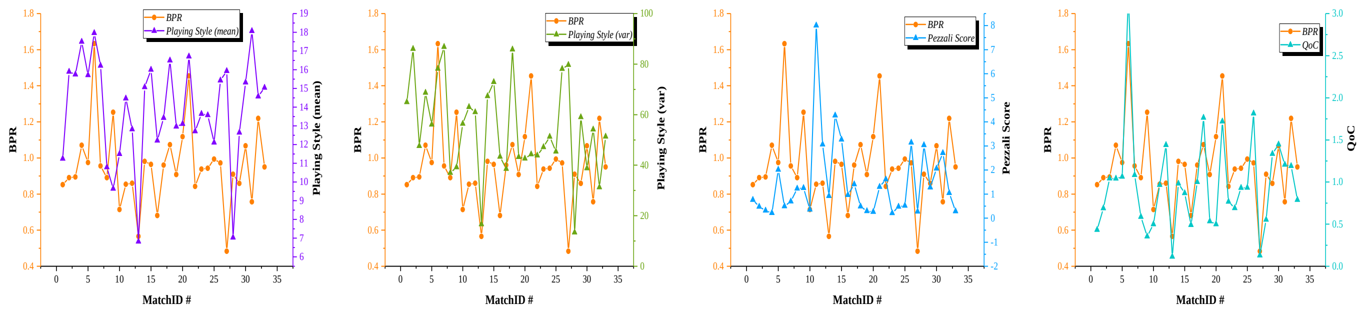


Fig. 8. QoC at different time stages of a game.

C. A Case Study of Edge Cooperation Optimization

In order to demonstrate the effectiveness of our proposed edge cooperation optimization method, we select a typical competition and analyze the changes of QoC. A game is



(a) Playing Style (mean) and BPR (b) Playing Style (var) and BPR (c) Pezzali Score and BPR (d) QoC and BPR

Fig. 7. Comparison of Quality of Cooperation (QoC) and baseline indicators with Ball Possession Rate (BPR) trends.

divided into 9 time stages, each of which contains 60 passes. The substitution event occurs between stage 4 and stage 5, and this decision is not consistent with the results of our proposed cooperative optimization algorithm. As a result, the QoC decreases correspondingly. As shown in the dotted box in Fig. 8. On the contrary, the substitution that occurs in stage 7 is consistent with our recommendations, and the QoC increases, which is marked by the solid box. According to the analysis of this case, experiments demonstrate that our proposed edge cooperation optimization method can timely evaluate ECN and provide feedback on the substitution events of the cooperative relationship, which can improve the scheme in ECN.

VI. Conclusion

In this paper, we develop an optimization framework for ECN named ECN-Opt. We first put forward the definition of the ECN and extract cooperative modes based on network representation learning. According to the preference of cooperative mode of ECN, we propose pervasive metrics QoC to evaluate the performance of cooperation in ECN. In addition, we design a cooperative optimization algorithm to improve the performance of ECN. In the experimental part, first, based on the dataset from wearable sensors on the players, we prove the effectiveness of our framework. Then, we conduct a case study to verify the feasibility of edge cooperative optimization algorithm.

References

[1] X. Kong, X. Liu, B. Jedari, M. Li, L. Wan, and F. Xia, "Mobile crowdsourcing in smart cities: Technologies, applications, and future challenges," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8095–8113, 2019.

[2] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.

[3] S. Saraswat, H. P. Gupta, T. Dutta, and S. K. Das, "Energy efficient data forwarding scheme in fog based ubiquitous system with deadline constraints," *IEEE Transactions on Network and Service Management*, vol. 17, no. 1, pp. 213–226, 2020.

[4] Z. Ning, K. Zhang, X. Wang, M. S. Obaidat, L. Guo, X. Hu, B. Hu, Y. Guo, B. Sadoun, and R. Y. K. Kwok, "Joint computing and caching in 5g-envisioned internet of vehicles: A deep reinforcement learning-based traffic control system," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2020.

[5] X. Kong, F. Xia, J. Li, M. Hou, M. Li, and Y. Xiang, "A shared bus profiling scheme for smart cities based on heterogeneous mobile crowdsourced data," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1436–1444, 2020.

[6] Z. Ning, K. Zhang, X. Wang, L. Guo, X. Hu, J. Huang, B. Hu, and R. Y. K. Kwok, "Intelligent edge computing in internet of vehicles: A joint computation offloading and caching solution," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2020.

[7] Z. Qu, B. Ye, B. Tang, S. Guo, S. Lu, and W. Zhuang, "Cooperative caching for multiple bitrate videos in small cell edges," *IEEE Transactions on Mobile Computing*, vol. 19, no. 2, pp. 288–299, 2020.

[8] C. Funai, C. Tapparello, and W. Heinzelman, "Computational offloading for energy constrained devices in multi-hop cooperative networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 1, pp. 60–73, 2019.

[9] T. Decroos, L. Bransen, J. Van Haaren, and J. Davis, "Actions speak louder than goals: Valuing player actions in soccer," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1851–1861.

[10] G. Liu and O. Schulte, "Deep reinforcement learning in ice hockey for context-aware player evaluation," *arXiv preprint arXiv:1805.11088*, 2018.

[11] L. Bransen, J. Van Haaren, and M. van de Velden, "Measuring soccer players' contributions to chance creation by valuing their passes," *Journal of Quantitative Analysis in Sports*, vol. 15, no. 2, pp. 97–116, 2019.

[12] J. Fernández, L. Bornn, and D. Cervone, "Decomposing the immeasurable sport: A deep learning expected possession value framework for soccer," in *13th MIT Sloan Sports Analytics Conference*, 2019.

[13] H. R. E. H. Bouchekara, "Most valuable player algorithm: a novel optimization algorithm inspired from sport," *Operational Research*, vol. 20, pp. 139–195, 2020.

[14] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Y. Chang, "Network representation learning with rich text information," in *Proceedings of the 24th International Conference on Artificial Intelligence, IJCAIâĂŹ15*, 2015, pp. 2111–2117.

[15] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 2014, pp. 701–710.

[16] S. Chang, W. Han, J. Tang, G.-J. Qi, C. C. Aggarwal, and T. S. Huang, "Heterogeneous network embedding via deep architectures," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 119–128.

[17] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[18] D. Mascitti, M. Conti, A. Passarella, L. Ricci, , and S. K. Das, "Service provisioning in mobile environments through opportunistic computing," *IEEE Transactions on Mobile Computing*, vol. 17, no. 12), pp. 2898–2911, 2018.

[19] Z. Q. Jaber and M. I. Younis, "Design and implementation of real time face recognition system (rtfrs)," *International Journal of Computer Applications*, vol. 94, no. 12, pp. 15–22, 2014.

[20] X. Luo, "From augmented reality to augmented computing: A look at cloud-mobile convergence," in *2009 International Symposium on Ubiquitous Virtual Reality*, 2009, pp. 29–32.

[21] A. Thiagarajan, L. Ravindranath, H. Balakrishnan, S. Madden, and L. Girod, "Accurate, low-energy trajectory mapping for mobile devices," in *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*, 2011, p. 267âĂŞ280.

[22] M. Yan, B. Chen, G. Feng, and S. Qin, "Federated cooperation and augmentation for power allocation in decentralized wireless networks," *IEEE Access*, vol. 8, pp. 48 088–48 100, 2020.

[23] I. Leyva-Mayorga, R. Torre, V. Pla, S. Pandi, G. T. Nguyen, J. Martinez-Bauset, and F. H. P. Fitzek, "Network-coded cooperation and multi-connectivity for massive content delivery," *IEEE Access*, vol. 8, pp. 15 656–15 672, 2020.

[24] F. Cheng, T. Chen, and Q. Chen, "Rewards based on public loyalty program promote cooperation in public goods game," *Applied Mathematics and Computation*, vol. 378, 2020.

[25] P. Cintia, F. Giannotti, L. Pappalardo, D. Pedreschi, and M. Malvaldi, "The harsh rule of the goals: Data-driven performance indicators for football teams," in *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2015, pp. 1–10.

[26] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 2016, pp. 855–864.

[27] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2015, pp. 1067–1077.

[28] H. Zhu, F. Feng, X. He, X. Wang, Y. Li, K. Zheng, and Y. Zhang, "Bilinear graph neural network with node interactions," *arXiv preprint arXiv:2002.03575*, 2020.

[29] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," *arXiv preprint arXiv:2002.02126*, 2020.

[30] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.